A novel approach to Bluetooth and GPS based contact tracing using a Health Application

Guess Who I Am
International Institute of Information Technology
Hyderabad, India
guesswho.iam@research.iiit.ac.in

Abstract

With the Sars-Cov 2 pandemic on its decline in many countries, governments are keen to restart businesses, but some still fear of a second wave of cases. In other countries where the cases are still on the rise, governments are scrambling to trace diagnosed residents to other potentially infected individuals. Traditionally, such a task is done manually, but due to the sheer size of the countries and the expensive nature of manual tracing, most countries cannot afford it. To alleviate this issue, I propose a novel method that automates tracing diagnosed patients to other participating individuals by using Bluetooth (Low Energy) and GPS technology of their phones to acquire the proximity of the participating individual to the patient, and subsequently alert the individual of the contact through a health app. The health app could be provided by the government. It could additionally be used to assign risk of infection and prioritize individuals for testing.

1. Introduction

Contact tracing has been the key to success for many countries to flatten the curve and restart businesses much sooner with little risk for a second wave developing.

1.1. Previous Cases of Contact Tracing

Wired [4] has reported the performance of contact tracing in South Korea. South Korea as of March 18th reported an average of 90 cases daily – a huge drop from whopping 900 cases daily just 2 weeks before. South Korea has now emerged victorious against the pandemic reporting only approx. 10 cases [8] daily even while having all businesses open.

Contact tracing was even used in 2014-2015 to cease the Ebola pandemic in Liberia [18]. Though limited by organizational issues and community mistrust, this technique saw great improvements in containing the spread of the virus.

1.2. Successful current contact tracing attempts

Manual contact tracing is a very tedious and expensive task. It takes 5-6 days to completely trace the movements of a diagnosed patient. South Korea has used manual tracing [9] along with a public map to warn the residents and provide the last locations of the infected patient, which has led to the fastest containment of the virus among affected countries. However, this method is not scalable to regions with higher population. Moreover, this kind of system has led to public harassment, boycotts and even tracking of infected patients, putting people in huge risks [16].

China [11] and Singapore [10] governments have released apps that automated contact tracing which has helped flatten the curve for the country. These systems are intrusive on the personal privacy as the complete data of all the individuals are sent to the Government. This approach may spur high level surveillance of the residents long after the pandemic is ceased [13] [16].

1.3. Basic Principle of the Proposed Software System

Taking inspiration from the success of automated systems. I would like to propose a novel system that abides by the following principles to ensure personal privacy:

1. The software system collects the minimal data.
2. The software system lets the complete consent on the user.
3. The software system should be dismantlable after the pandemic is over.
4. The software system may not be employed for any malicious use.
5. The software system values personal privacy more than the general benefit of others.

With the great performance of automated contact tracing in many countries, a more privacy-centric contact tracing

application is the need of the hour. Not only will it encourage a higher adoption but also prevent the misuse of such technology in the future.

2. Literature Review

Many approaches have been taken to build an automated contact tracing application. Some of them employ Bluetooth Technology, others employ GPS technology.

2.1. GPS Technology based contact tracing

MIT's SafePath [15] Kit focuses on tracing contact with an infected individual through the Location Trails of that person. The idea is that the application locally stores the locations (through GPS) visited by a person in the past 14 days. If that person is diagnosed with COVID-19, he/she will have the ability to make his/her location data public to the participants of the application. All the participants may download the encrypted location of the diagnosed person and compare to determine whether they have crossed paths with that person in the past 14 days.

2.2. Bluetooth Technology based contact tracing

While most of the initial designs were based on GPS technology to detect location, most of the current implementations rely on Bluetooth technology to detect proximity. This idea has been spear-headed by Apple and Google who are planning to release APIs [2] (Application Programming Interface) to allow developers to make apps based on Bluetooth tracing.

Pan-European Privacy-Preserving Proximity Tracing [14] (and Covid Watch [3]) has developed Bluetooth technology-based proximity tracing mobile applications that help trace individuals to patients. This is among many applications that have been developed to use such technology; however, the security standards of these applications are still questionable. The application developed by PEPP-PT has been warned [20] of not meeting the security standards.

2.3. Concerns with current implementations

A common issue of all these applications is that Bluetooth technology is not completely reliable at detecting the strength and therefore measuring the distances of the devices. Another concern is that a device by itself may not be able to detect all the individuals around it. This could be due to the alignment of the device or the placement of the device [12]. My method attempts to solve this by incorporating results of network devices that are closest to the broadcasting device through a consensus algorithm to confirm individuals who are in proximity of the broadcasting device.

For location sharing aspect of the application, I propose to use a hash function with a salt that periodically changes. This would prevent the actual co-ordinates from being recovered from the data received, as both the location broadcaster and the individual in contact of the broadcaster would only have the hashed values of the locations. This would prevent boycotting and harassment of local businesses.

2.4. Other techniques of contact tracing

CoEpi [5] is a web open platform (developed by the open source organization CoEpi) which allows a diagnosed person to publish the locations that he visited in the past 14 days. The trouble with such an approach is that it leads to a large number of false positives. Furthermore, it creates unnecessary panic among the people and establishes a stigma towards certain locations and businesses.

3. System Architecture

The entire system is broken down into 3 major components:

1. Bluetooth based proximity tracing
2. GPS based location tracing
3. Health Application to model the probability of infection

The health application is one that could be provided by the government and would mandatorily take basic details and pre-existing medical conditions of the person installing the application. This system could then just use this information to model the vulnerability of that individual being infected. The remaining two components are not mandatory and are activated only through the consent of the user. The data from the remaining two components may be fed into the health application's model to improve the probability-risk calculation, and to prioritize individuals.

3.1. Bluetooth Based Tracing

This component uses Bluetooth Low Energy protocols to record the proximity of another participant.

3.1.1 Working

In a basic model each user is assigned a periodically changing unique temporary contact number (TCN) which serves as a key for that user. The periodic nature of the TCN is to prevent users from being identified or tracked. The period of change may vary based on the implementation, however, according to [19] 15 minutes is a good period to keep. The length of the TCN may again vary based on the implementation. [19] suggests using at least a ten-bytes key to prevent collisions, while [20] uses anywhere from 28-64 bytes keys. [20] prefer to generate TCN through hashing of

a periodically changing id through a pseudo-random function, while a simple generation without the use of hashing but only random sequence of numbers may suffice if the key is long enough. It is again left to the implementation if whether the set of keys should be generated at once (say every morning) or should be generated every 15 mins, but [20] suggests generating a set of keys at the start of the day.

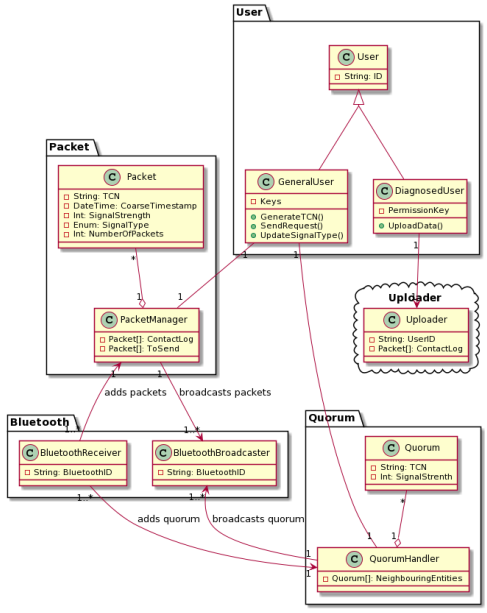


Figure 1: Class Diagram of the Bluetooth component of the proposal

When two users, say Alice and Bob, come in proximity of one another, they share their contact numbers with one another. Both then maintain a contact log of the exchanged numbers. The contact log is stored locally on the devices of the participant and are not sent to any third-party agent without the consent of the user. Along with the numbers other details are also shared:

- Coarse Timestamp (like 28th April)
- Strength of Signal
- Status of Signal
- Number of messages exchanged

Detail 3 is updated later as more information arrives to the devices, while detail 4 is constantly increased as both Alice and Bob spend more time in proximity of one another.

Once, say Bob, is diagnosed positive with COVID-19, he will be given an active permission key from the government. He can then use that permission key on his own consent to upload his keys (or contact logs) for past 14 days to a common public database. The details are deleted from the database 14 days after it was uploaded.

Alice can check on her device and download the keys of diagnosed patients daily. Since she was in contact with Bob, for a duration greater than a specified threshold, her application will warn her that she may be potentially infected. To identify Bob, she will recompute the Bob's hashes from the keys released that were employed to generate the TCN. She may now be able to share that specific contact log with the health application which will provide her with a probability of infection and better risk assessment.

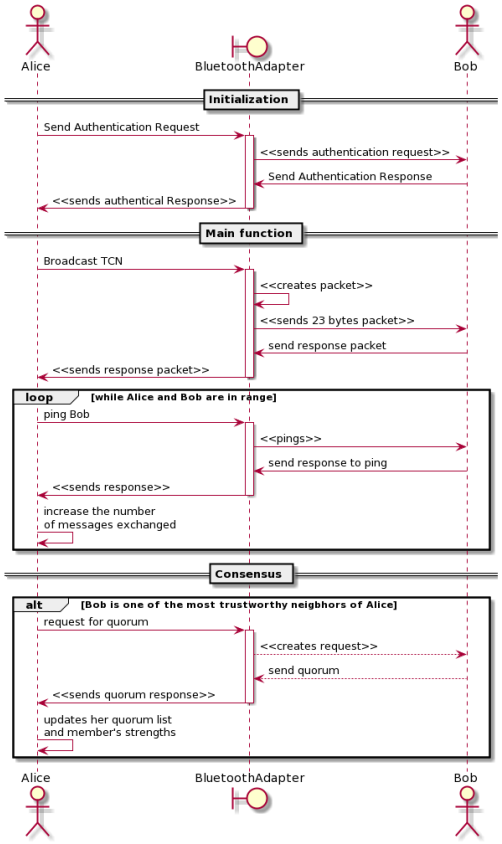


Figure 2: Sequence Diagram of the Bluetooth component of the proposal

3.1.2 Consensus Algorithm

The novel consensus algorithm is inspired from algorithm known as Federated Byzantine Agreement (FBA) [17]. In FBA, each participant creates a quorum of other participants that he/she feels is valid. Now once enough number of quorums are formed. The participants look at the intersection of all the values. The ones with the highest intersecting are found to be the most trustworthy. From here the original algorithm deviates from the proposed one

as it goes on to deal with the unanimous agreement for blockchains.

The proposed algorithm is as follows. Once a device has received all the responses to its initial packet broadcast, it sends out another broadcast to two (or more) of its most trustworthy neighboring devices. The trust may be calculated based on weighted mean of the short-term exponential average of signal strength and the number of messages exchanged with that device. In that broadcast, the main device requests its neighbors to send their quorum of top k nearest (or most trustworthy) devices. Once the main device receives the responses from its neighbors, it estimates what strength would it perceive of the members of the neighbors' quorums. It then compares the estimated result with what it has measured for those devices, and either retains its original measurement or modifies it based on the majority. However, if no consensus can be reached on the strength (distance) of the other devices, it keeps it unconfirmed. The main device then assigns either one of the 3 types to the detail 3:

1. Near and confirmed
2. Far and confirmed
3. Unconfirmed

3.1.3 Data Transmission

BLE was introduced in Bluetooth 4.0 [1]. While some devices still have 4.0 most of them have either moved on to 4.2 or 5.0.

BLE, out of the box, allows 23 bytes of packets to be exchanged between devices [6]. It is possible to design a data packet that would fit completely within the 23 bytes:

Data	Bytes Required
TCN	12 bytes
Timestamp	4 bytes
Signal Strength	2 bytes
Signal Type	1 byte
Number of messages exchanged	2 bytes

Table 1: Sections of the packet with their length requirement

The additional 2 bytes are left for the Bluetooth exchange protocol keys.

However, to discover neighbor's quorum a larger amount of data transfer will be required. For this the MTU (maximum transmission unit) protocol will be invoked to increase the packet length to the maximum of 512 bytes [6]. The packets will then contain the TCN of the member of the quorum and the associated signal strengths. This will allow the user to share details of up to 36 members of its quorum, which in current scenario seems adequate. If this falls short, a secondary request may also be sent to get more results, but this may lead to slow speed of updating details.

3.1.4 Privacy

To ensure user privacy all the data will be stored locally on the device unless the person decides to make it public given the permission to do so.

Besides this to ensure that the Bluetooth data transfer cannot be read or influenced by malicious actors I propose using a secret sharing scheme to share the data. A k out of n sharing scheme [20] or Shamir's Sharing Scheme [21] will suffice for the application. This will ensure that a malicious actor will not be able to get all the details instantly. At the same time, this will require more requests to be transferred between the parties before recognition which will add an additional threshold on the amount of time that has to be spent together to begin sharing details, thus reducing the number of false positive cases.

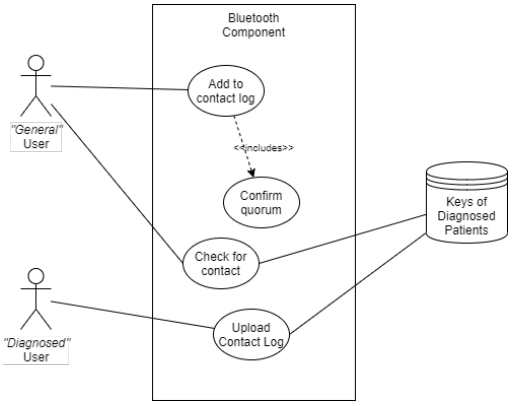


Figure 3: Use Case Diagram of the Bluetooth component of the proposal

3.2. Location Based Tracing

A method similar to [16] is suggested. The location details of the user will be stored locally.

3.2.1 Location Hashing

The modification I propose is that the details of the GPS location will be hashed with a periodic salt that is common to all the users. The period of the key generation may vary on implementation. I suggest a period of 1 – 2 hours to prevent malicious actors from identifying locations. The periodic nature of the salt may also simply be used to note that participating users have visited the location within the timeframe.

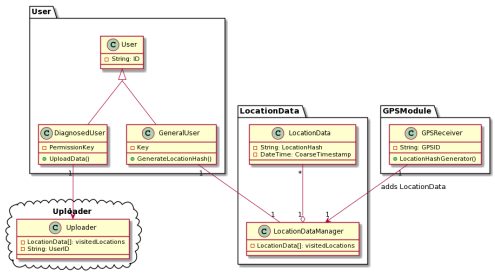


Figure 4: Class Diagram of the Location-based component of the proposal

3.2.2 Working

In this component, as a participating user goes about his daily work, his GPS location will be fetched once every few minutes and stored on the device locally. This location data will be generated only using handheld hardware module for GPS, and not any other method (like Cell Tower Triangulation) in order to preserve the privacy of the user. Furthermore, the location co-ordinates will be hashed by the application and then stored into the device.

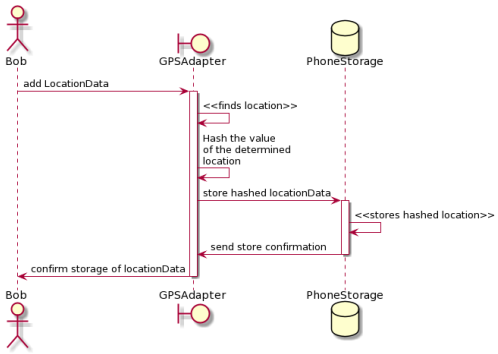


Figure 5: Sequnce Diagram of the Location-based component of the proposal

Assume that Bob and Alice have crossed paths within the timeframe, and the location details are stored locally on both the phones. Say Bob is diagnosed positive with COVID-19. Now with his consent his encrypted and hashed location details are made available to participants who can now download that data and compare. Since Alice had crossed paths with Bob, she will be warned as potentially infected.

3.3. Health App

This forms the brains of the operation and is responsible for alerting the individuals. The health application can be distributed by the government.

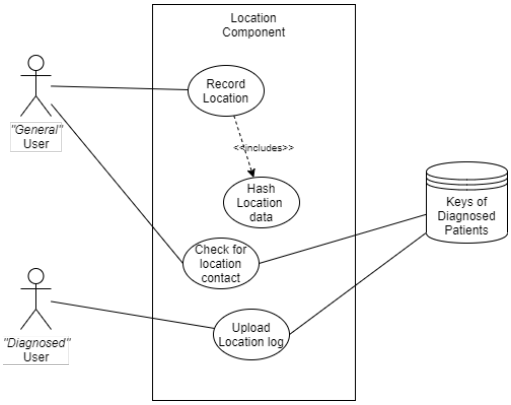


Figure 6: Use Case Diagram of the Location-based component of the proposal

3.3.1 User Front-end

As a general user, a person will be able to enter his medical records (if needed) or pre-existing health conditions along with some basic details. The application will also perform risk analysis using various models that can be developed using various details. The application will also show announcements from the government and provide tips for maintaining good health and best practices. The user will also be able to upload his contact logs (and/or location details) to the database either for publishing or for checking for contact with a diagnosed person. The application will also feature a notification system to send an emergency alert in case of confirmed contact. The application will also feature an online medical support team that will comfort and advise the individuals.

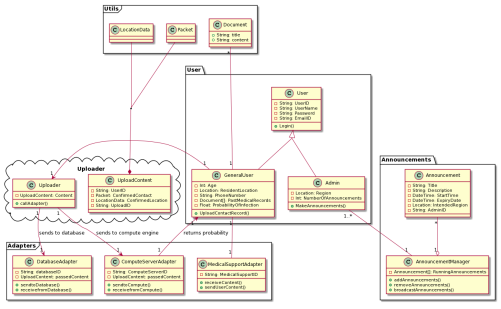
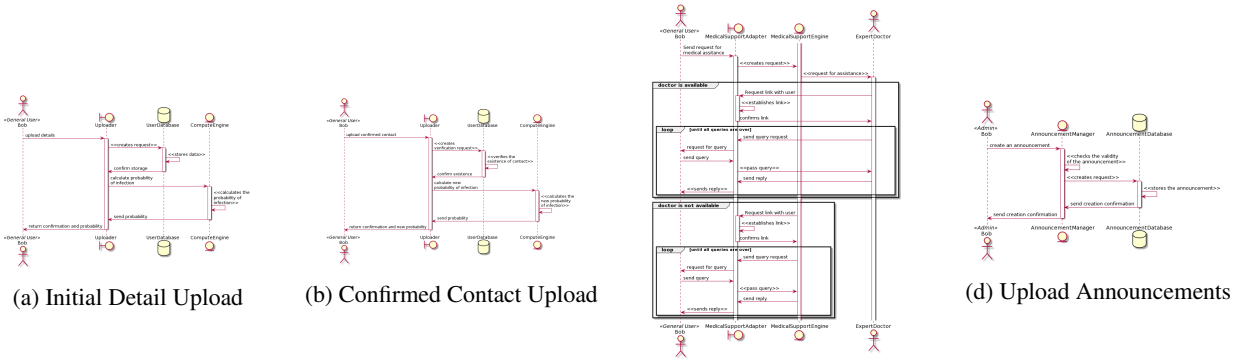


Figure 7: Class Diagram of the Application component of the proposal

The application should have the following basic requirements:

1. A page for the user to register and enter his basic details



(a) Initial Detail Upload

(b) Confirmed Contact Upload

(c) Medical Support

(d) Upload Announcements

Figure 8: Sequence Diagrams of Various Functions of the App Component of the proposal

2. A page for the user to enter past medical records and pre-existing conditions
3. A page for the user to upload his contact log details for verification
4. A page for the user to receive permission keys
5. A page for the user to upload his/her records using the permission keys
6. A page to display government health announcements
7. A page for the user to export his details
8. The application should be minimal and must ask for only details that are essential but not intrusive.
9. The application should also allow the users with their consent to share their data anonymously with epidemiologists to improve further predictions.

3.3.2 Backend

The data transmission from the frontend to the backend servers must be done through encrypted protocols. HTTPs may be a good example of a communication protocol that can be employed here. The backend server has access to two other components and deals with the interaction of the mobile application with those two components:

1. Compute Engine / Compute Server
2. Database

3.4. Compute Server

The compute server will help for filtering out, prioritizing cases and for performing risk analysis. Singapore and China have implemented a filtering process for the cases into their applications, however, little detail is provided about the filtering process except for a crude threshold of Bluetooth strength.

My proposal will vastly improve that as we may be able to train Deep Learning Models to predict with higher accuracy. The inputs to such a model may be:

- Health details
- Location of user
- Epidemiological Results
- Proximity to diagnosed patient

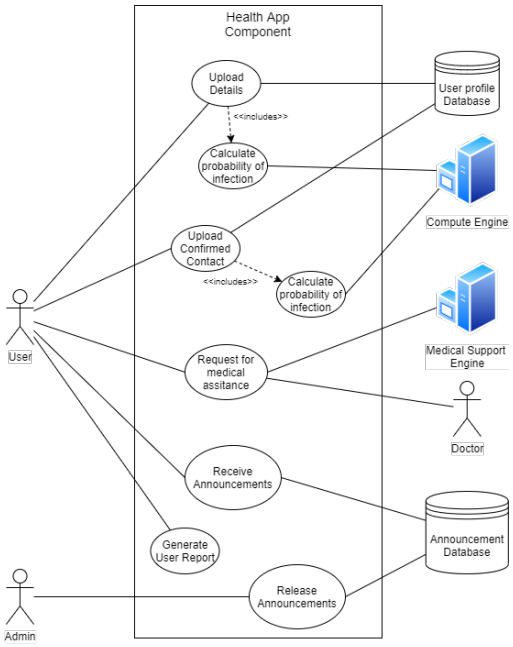
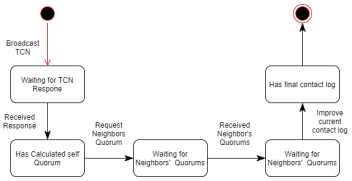
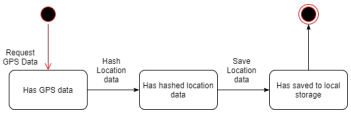


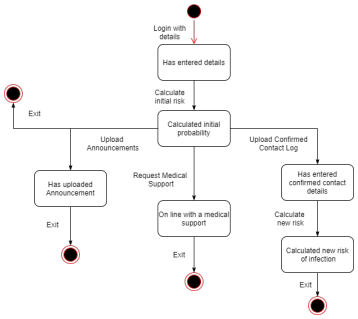
Figure 9: Use Case Diagram of the Application component of the proposal



(a) Bluetooth-based Component



(b) Location-based Component



(c) Application Component

Figure 10: State Diagrams of the components of the proposal

- Duration of contact
- Signal Type
- Density of contact location which could be calculated using diagnosed patient's contact log.
- Other results such as NBDA

Initially, since the dataset may be small leading to less confident predictions, however the system allows for more data to be published through the application which will lead to more confident results in the future.

The compute server of the application will be handled by epidemiologists and medical researchers to develop models that will incorporate the user details along with various other details such as social transmission probability, density of social networks, NBDA, etc.

3.4.1 Pipeline

Once the user uploads his confirmed contact log, his data will be fed into a model that will attempt to predict the risk of infection for that person. The user's details may also be combined with other details such as location details, spread rate details, etc.

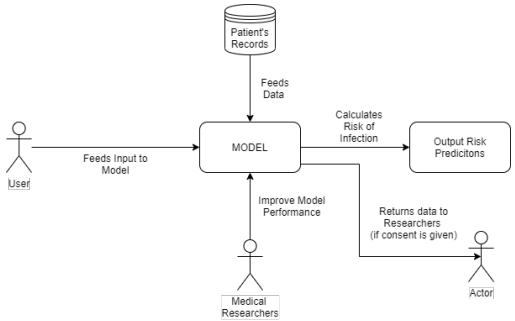


Figure 11: Flow of the Compute Engine

A government platform can be developed associated with the compute server that would allow various medical institutes to receive the anonymous patient data as well as feed their findings into the system. The inputs from the institutes may be review before being incorporated with the exact model.

The design of the model and inputs and outputs may vary based on the implementation by the developers.

3.5. Databases

In the model proposed there are 3 different databases:

1. Key of diagnosed patients for Bluetooth component
2. Key of diagnosed patients for Location component
3. User profile details for the health application

From my calculation the storage requirements for the first two components will be well within limits. The following show the calculation for the Bluetooth component.

We store 12 bytes of the TCN which change 4 times every hour. Now considering a conservative average of 2000 cases daily and a record period of 14 days we get:

$$\text{data} = 12 * 24 * 4 * 2000 * 14$$
$$\text{data} = 32,256,000 \text{ bytes}$$
$$\text{data} \approx 33 \text{ Mb}$$

which is well within the storage limits of modern day servers.

3.5.1 Pipeline

The database will be handled by many users of various roles. A general user may use it to download the stored details of diagnosed patients. A diagnosed patient may upload his records to the database. Epidemiologists may get data from the database upon user consent to develop new models. Doctors may upload details of diagnosed patients, which will be used by the model to improve predictions.

3.6. A discussion of the design concepts of the proposal

The proposal shows the following design concepts:

- Abstraction: The components show abstraction as it hides the unnecessary details of the components from the user. The user is shown only the important details of each component.
- Separation of Concerns: In the proposal, we can see from the UML diagrams that many multi-stepped processes have been separated into combinations of simpler ones which are done by different parts of the code. We can see this in separation of the task of making the contact log, where the process is divided into getting the initial data and improving the data.
- Modularity: This is a strong feature of the proposal as the main task of tracing and assigning risk to an individual has been divided into various modules – Bluetooth module, location-based module, application frontend module, compute engine (model for risk analysis) module.
- Hiding: As mentioned in the abstraction section, the client has no idea of the internal model and working of the machinery. This is apparent in the application section as the user only enters his details and is unaware of the model that may run on that to perform risk assessment.
- Each module of the proposed design shows high cohesion as they are highly internally related, while the module also shows low coupling with the other modules, as the only data that is transferred across the modules is the contact log.

3.7. Conclusion

Detailed work, analysis and scrutiny of each steps have been done rigorously before concluding the following proposals:

1. A novel method for improving the accuracy of Bluetooth contact tracing through use of Consensus algorithm.
2. A novel method for improving the privacy of location-based contact tracing
3. A proposal to utilize data to train models to assign probabilities of infection to each user, whether they were in contact of a diagnosed patient or not, through the use of related data from medical institutions.
4. A novel application proposal that allows Government intervention at restricted level to provide a middleware to access database data as well provide filter patients through deep learning models discussed in proposal 3.
5. The current proposal combines current detection technology with mobile applications to further improve predictions and prioritizations.

The main focus of the paper was to discuss the current methods of contact tracing and suggest new and slight improvements to those methods to improve the accuracy in terms of lower false positives and false negatives, as well as improve the privacy of the methods.

3.8. Future Work

In a future work, the Bluetooth based contact tracing could be improved to incorporate Bluetooth 5.0 features to detect the direction of the signals [7].

Another focus of a future work could be the online medical support system. The model currently suggests using an AI based support team in the absence of an expert doctor. This could be further expanded with greater developments in those fields.

References

[1] M. Afaneh. Bluetooth low energy: A primer, 2019. 4

[2] Apple and Google. Apple and google partner on covid-19 contact tracing technology, 2020. 2

[3] S. V. Arx et al. Covidwatch, 2020. 2

[4] W. Bedingfield. What the world can learn from south korea's coronavirus strategy, 2020. 1

[5] CoEpi. Coepi: Community epidemiology in action, 2020. 2

[6] C. Coleman. A practical guide to ble throughput, 2019. 4

[7] D.-K. N. A. Editors. Use bluetooth 5.1-enabled platforms for precise asset tracking and indoor positioning. Technical report, Digi-Key Electronics, 2019. 8

[8] KCDC. Coronavirus infection-19 domestic outbreak status (april 30, regular briefing). Technical report, KCDC, 2020. 1

[9] M. Ketchell. South korea's success in controlling disease is due to its acceptance of surveillance, 2020. 1

[10] D. Koh. Singapore government launches new app for contact tracing to combat spread of covid-19, 2020. 1

[11] E. Law. Coronavirus: China's contact tracing app touted as helping to contain outbreak, 2020. 1

[12] P. H. O'Neill. Bluetooth contact tracing needs bigger, better data, 2020. 2

[13] D. Palmer. Coronavirus contact-tracing apps: What are the privacy concerns?, 2020. 1

[14] pepp pt. Pan-european privacy-preserving proximity tracing, 2020. 2

[15] M. M. Ramesh Raskar et al. Private kit: Safe paths, 2020. 2

[16] R. Raskar, I. Schunemann, et al. Apps gone rogue: Maintaining personal privacy in an epidemic. *ArXiv e-print*, 2020. 1, 4

[17] S. Ray. Federated byzantine agreement, 2018. 3

[18] K. C. Swanson, C. Altare, C. S. Wesseh, T. Nyenswah, T. Ahmed, N. Eyal, E. L. Hamblion, J. Lessler, D. H. Peters, and M. Altmann. Contact tracing performance during the ebola epidemic in liberia, 2014-2015. *PLOS*, 2018. 1

[19] team-covid-19-contact-discovery project. A simple proximity-based approach to contact tracing. *pre-print*, 2020. 2

[20] P. C. Troncoso et al. Decentralized privacy-preserving proximity tracing, 2020. 2, 3, 4

[21] Wikipedia. Shamir's secret sharing, 2020. 4